Implementing a Data Mining Algorithm

CS 4378U - Introduction to Data Mining – Spring 2010

Demo date: end of semester, TBA on course webpage

Objective: Implement a data mining algorithm of your own choice and experimentally demonstrate its correctness and efficiency.

Team size: 1

General Description:

Assume the case that we are building a repository of data mining algorithms for free public access. Try to make your program easy to use.

You choice is NOT subject to approval. However, if the algorithm is too difficult, you may not be able to finish it. If it is too simple, you may not receive good feedback and evaluation in demo. So you may want to consult the instructor about your choice.

Tasks:

1. Understand the algorithm and implement it using any programming language you prefer. Concern correctness, efficiency and usability in your implementation.
2. Design and perform experiments to show the correctness and efficiency of your implementation. For this purpose, you may need to do some research, e.g., read the original paper or some follow-up papers and see how they set up the experiments.
3. Write a report. In roughly 5 pages (no lower or upper page limit), introduce the algorithm, describe your implementation, report your experiments, and show how to use your program (user manual). Make references properly in your report.

Example Choices of Algorithms:

1. Some well-know clustering algorithm, such as DBSCAN, BIRCH, ROCK, CHAMELEON, CLIQUE, EM, pCluster, etc.
2. Some well-know classification algorithm.
3. Some well-known pattern mining or sequential pattern mining algorithm.
4. Some well-known information retrieval and web search algorithm, such as the Rocchio algorithm for (pseudo) relevance feedback, HITS or PageRank for web search. In particular, it is interesting to implement a PageRank demo, where

through a web interface, users can draw a toy web graph (nodes representing pages and edges representing hyperlinks) and calculate the PageRank values.

Note: it is greatly encouraged that your choice of algorithm can be application-driven. For example, some students are interested in text clustering or text classification. Then, go ahead to do some research and find/adapt/design some appropriate algorithm and conduct some experiments. Feel free to consult the instructor in this process.

Evaluation: You will have 5 – 15 minutes to demonstrate your work.

The demo will be evaluated by peer students and the instructor. Projects will be ranked. A weighted (peer students 0.5 in total, instructor 0.5) average ranking will be calculated, based on which, 0 ~ 100 points will be assigned to each team by the instructor. Detailed scheme will be given at the demo. Note that although nice presentations help, presentation skills should be not the focus for this evaluation.

Also, your evaluation will be evaluated based on the correlation coefficient between your ranking and the average ranking. **0 ~ 5** bonus points will be added to your project, which, however, should not bring your total points for the project beyond 100.

Exceptional projects will be specially evaluated. You may be considered for a happy course grade despite your performance elsewhere.

Submission: Zip your source code, executable, sample datasets if any, and report in a single file, submit to TRACS before the demo.